Multidimensional Data Display by Nonlinear Mapping

V. Zitko

Biological Station,
St. Andrews, N. B., E0G 2X0

January 1986

Canadian Technical Report of Fisheries and Aquatic Sciences No. 1428

86-03423/

Canadian Technical Report of Fisheries and Aquatic Sciences

Technical reports contain scientific and technical information that contributes to existing knowledge but which is not normally appropriate for primary literature. Technical reports are directed primarily toward a worldwide audience and have an international distribution. No restriction is placed on subject matter and the series reflects the broad interests and policies of the Department of Fisheries and Oceans, namely, fisheries and aquatic sciences.

Technical reports may be cited as full publications. The correct citation appears above the abstract of each report. Each report is abstracted in *Aquatic Sciences and Fisheries Abstracts* and indexed in the Department's annual index to scientific and technical publications.

Numbers 1 456 in this series were issued as Technical Reports of the Fisheries Research Board of Canada. Numbers 457 714 were issued as Department of the Environment, Fisheries and Marine Service, Research and Development Directorate Technical Reports. Numbers 715 924 were issued as Department of Fisheries and the Environment, Fisheries and Marine Service Technical Reports. The current series name was changed with report number 925.

Technical reports are produced regionally but are numbered nationally. Requests for individual reports will be filled by the issuing establishment listed on the front cover and title page. Out-of-stock reports will be supplied for a fee by commercial agents.

Rapport technique canadien des sciences halieutiques et aquatiques

Les rapports techniques contiennent des renseignements scientifiques et techniques qui constituent une contribution aux connaissances actuelles, mais qui ne sont pas normalement appropriés pour la publication dans un journal scientifique. Les rapports techniques sont destinés essentiellement à un public international et ils sont distribués à cet échelon. Il n'y a aucune restriction quant au sujet; de fait, la série reflète la vaste gamme des intérêts et des politiques du ministère des Pêches et des Océans, c'est-à-dire les sciences halieutiques et aquatiques.

Les rapports techniques peuvent être cités comme des publications complètes. Le titre exact paraît au-dessus du résumé de chaque rapport. Les rapports techniques sont résumés dans la revue *Résumés des sciences aquatiques et halieutiques*, et ils sont classés dans l'index annual des publications scientifiques et techniques du Ministère.

Les numéros 1 à 456 de cette série ont été publiés à titre de rapports techniques de l'Office des recherches sur les pêcheries du Canada. Les numéros 457 à 714 sont parus à titre de rapports techniques de la Direction générale de la recherche et du développement, Service des pêches et de la mer, ministère de l'Environnement. Les numéros 715 à 924 ont été publiés à titre de rapports techniques du Service des pêches et de la mer, ministère des Pêches et de l'Environnement. Le nom actuel de la série a été établi lors de la parution du numéro 925.

Les rapports techniques sont produits à l'échelon régional, mais numérotés à l'échelon national. Les demandes de rapports seront satisfaites par l'établissement auteur dont le nom figure sur la couverture et la page du titre. Les rapports épuisés seront fournis contre rétribution par des agents commerciaux.

Canadian Technical Report of
Fisheries and Aquatic Sciences 1428

January 1986

MULTIDIMENSIONAL DATA DISPLAY BY NONLINEAR MAPPING

by

V. Zitko

Fisheries and Environmental Sciences
Department of Fisheries and Oceans
Biological Station
St. Andrews, New Brunswick EOG 2XO
Canada

This is the one hundred and eighty-third Technical Report from
the Biological Station, St. Andrews, N. B.

© Minister of Supply and Services Canada 1986
Cat. No. Fs 97-6/1428E ISSN 0706-6457

Correct citation for this publication:

Zitko, V. 1986. Multidimensional data display by nonlinear mapping. Can. Tech. Rep. Fish. Aquat. Sci. 1428: iii + 10 p. + Appendix.

ABSTRACT

Zitko, V. 1986. Multidimensional data display by nonlinear mapping. Can. Tech. Rep. Fish. Aquat. Sci. 1428: iii + 10 p. + Appendix.

This report presents two nonlinear mapping programs. The programs map points from N-space to 2-space, preserving their distances as much as possible. The program OBRAZ performs the conventional nonlinear mapping for 100 points. The program SVETB performs this on 40 points and a simplified nonlinear mapping on any number of points. The programs are written in HP 3000 BASIC and the listings are presented in the appendix.

RÉSUMÉ

Zitko, V. 1986. Multidimensional data display by nonlinear mapping. Can. Tech. Rep. Fish. Aquat. Sci. 1428: iii + 10 p. + Appendix.

Ce rapport présente deux programmes de cartographie non linéaire. Les programmes portent sur des cartes des points, situés dans des espaces de N à 2 dimensions, en conservant autant que possible les distances qui les séparent. Le programme OBRAZ exécute la cartographie non linéaire classique pour 100 points. Le programme SVETB fait de même pour 40 points et exécute une cartographie non linéaire simplifiée pour un nombre illimité de points. Les programmes sont écrits en HP 3000 BASIC et les listes imprimées sont présentées en annexe.

INTRODUCTION

Relationships and patterns can be seen much faster and easier from graphs than from tables. As the number of dimensions increases, graphic presentation becomes cumbersome. We are restricted to 2-dimensional plots of pairs of variables and to projections of multidimensional graphs into 2 or 3 dimensions.

When one measures "n" properties of a sample, the sample may be represented by a point in an n-dimensional space. If the values of these properties for another sample are similar, the samples are similar and the distance between the two points will be small. To visualize the distances between points in the n-dimensional space, one plots the points in a 2-dimensional space, trying to preserve their distances.

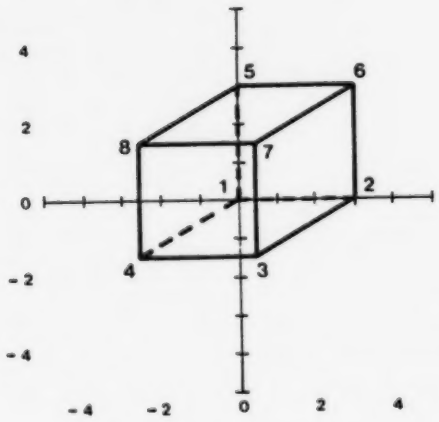


Fig.1 Projection of a cube
The projections of parallel edges are parallel but distances are distorted. The distance 1-7 is in reality the same as 4-6.

Fig.1 depicts a cube. Since we know what a cube looks like we know that the distance between the vertices 1 and 7 is larger than indicated in Fig.1. Actually, the distance between 1 and 7 is the same as the distance between 4 and 6. The projection does not preserve distances, but parallel lines have parallel projections, and using imagination and experience we obtain a correct perception of the three-dimensional relationships.

Distances are preserved by nonlinear mapping (NLM). Fig.2 is an NLM of a cube. Note that the distances 1-7 and 4-6 are approximately equal. The price to pay is that maps of parallel lines are not necessarily parallel. Consequently, the shape of objects becomes distorted. On the other hand, since we do not know what a multidimensional object looks like, the distortion is not important. The preservation of distances is, because we are interested in similarities, differences, and patterns, reflected by distances.

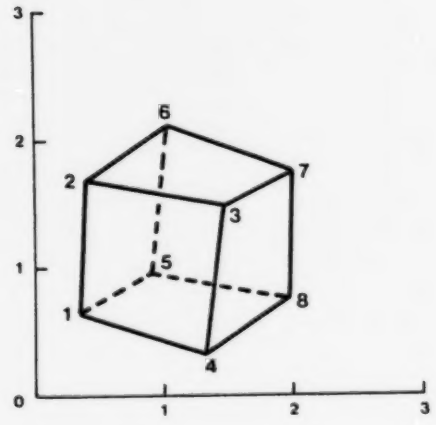


Fig.2. Nonlinear map of a cube.
Parallel edges are not mapped parallel, but the distances are fairly well preserved: the distances 1-7 and 4-6 are almost equal (compare with Fig.1).

NLM was introduced to chemistry by Kowalski and Bender (1972, 1973). For the role of display techniques in information extraction see Derde and Massart (1982) or Massart and Kaufmann (1983).

NLM programs are not available in the common statistical packages. The pattern recognition system ARTHUR (Anon 1985) contains a nonlinear mapping subroutine. Another program is available from the Tektronix library (Anderson 1981). The program OBRAZJ in this report is a translation of this program with input and output modified for convenient use with our current system (HP 3000) in a JOB mode.

There are N*(N+1)/2 distances between N points and it does not take too many points to run out of the work area of a computer. The computational effort also becomes exceedingly extensive, since many distances have to be recalculated as a result of each iteration.

A simplified approach was described by Forina et al (1983). A few points (3-10) are selected. These points may or may not be from the data set. They may be special points or could be selected at random. Their best distance-preserving 2-dimensional coordinates are calculated. Since the number of points is small, this is easy in terms of computations. Next, points to be mapped are taken individually and their 2-dimensional coordinates are calculated, preserving the distances to the

basepoints. The distances are weighted so that the basepoints closer to the point in N-space carry more weight. Since the points are mapped individually and in succession, there is no restriction of the number of points to be processed.

The price to pay is the loss of shape perception, as demonstrated in Figs.3-4 for a cube mapped when using different sets of basepoints. The distances of the points are preserved quite well.

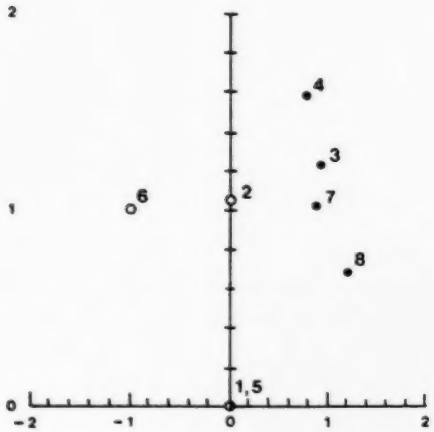


Fig.3. Simplified nonlinear map of a cube. Vertices 1,2, and 6 (see Fig.1) are basepoints (o). The remaining vertices are mapped to preserve their distances to the basepoints.

A program for this algorithm (Simplified Nonlinear Mapping, SNLM) is not readily available in the literature. The program SVETB can perform NLM on a maximum of 40 10-dimensional points or SNLM.

OBRAZJ

This is a translation of the program by Anderson (1981). It reads data from an input file and writes the mapped coordinates to an output file. The input data may be used as such, they may be centered (mean row subtracted), or centered and scaled (mean row subtraction and division by standard deviation).

The program minimizes the value of the error function E:

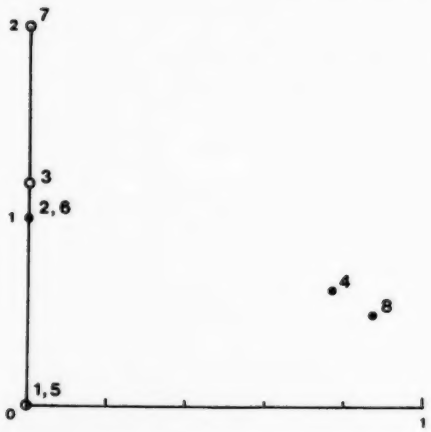


Fig.4. Simplified nonlinear map of a cube. Vertices 1,3, and 7 (see Fig.1) are basepoints (o). The remaining vertices are mapped to preserve their distances to the basepoints.

$$E = 1/SUM(d) * SUM((d - d^*)^2/d),$$

where SUM is summation for all distances between points i,j at i<j
d, d^* are the distances in N-space 2-space, respectively.

The minimization is performed by the "steepest descent" method for 40 iterations, with a fixed movement of 0.4 along the gradient in each step. No convergence checks are made.

The compiled version of the program occupies about 7K.

An example run is given below:

```
RUN OBRAZJC

Input filename CUBE
Output file CUBEF
        NONLINEAR MAPPING
The data matrix is
 0              0              0

 0              1              0

 1              1              0

 1              0              0

 0              0              1

 0              1              1
```

```
 1              1              1

 1              0              1

 Center (C) & scale (S) data (C,S/N) N
Mean
 .5             .5             .5

Variance
 .25            .25            .25

Mapping error  6.20788E-02
The 2-dimensional coordinates are
 .464521        .196669

 .330965        1.30363

 1.34127        1.25884

 1.61675        .258168

 .113543       -.301945

-.159172        .696836

 .994686        .757358

 1.12553       -.349429
```

The data from the output file may be used for plotting. The format is compatible with the plotting program AUTOPLOT available on our system.

SVETB

This program was written to perform NLM or SNLM. For the latter, up to 10 basepoints may be selected. The program accepts data from ASCII files or from the keyboard. As for OBRAZJ, the output file is compatible with the plotting program AUTOPLOT. A flowchart of the program is given in the Appendix. The compiled version of the program (SVETC) occupies about 13K.

In its usual version the program has space for 250 points in 10-space. The numbers of points and dimensions may be changed as long as one does not exceed 2500 real numbers. Unfortunately, this cannot be done interactively and the program must be modified as described in the documentation.

The program asks several questions:

Full printout (Y/N)

Normal response is "N". In the "Y" mode, the original data are displayed. It is a good idea to do this when one is not certain about the data being read correctly. In addition, the mapping error and 2-dimensional coordinates are displayed during each iteration.

Input manual(1), from file(2)

The usual situation is "2", from file, which leads to the next question:

Filename

The answer to this question provides some room for error. When the specified file does not exist, one may try again. This of course does not help during a JOB.

Center (C) & scale (S) data (C,S/N)

This deals with modifications of the input data. The data may be centered by subtracting the mean row (the mean of each column becomes zero). The data may also be scaled by dividing them by the square root of the sum of squares in each column (the vector length of each column becomes 1). Finally, the data may be left alone. This question is not asked when the number of data exceeds 250. In that case the data cannot be modified.

NLM (1), SNLM (2)

One may choose between NLM and SNLM. When the number of points exceeds 40, the program goes automatically to SNLM and this question is not asked. Keep it in mind when preparing job files.

For NLM the parameter input is now completed. For SNLM one still has to settle the input of basepoints.

Are basepoints in a special file (Y/N)

It is usually convenient to have basepoints in a file, in which case the program asks

Filename

The input of the basepoint file name is expected.

Basepoints from data file (1), manually (2)

This question is asked when there is no file of basepoints. Basepoints may be selected from the current data file and sections of the data file are displayed. The basepoints may also be entered manually.

This completes the opening dialogue for SNLM.

An example of a run is given below:

```
:RUN SVETC

Full printout (Y/N) N
Input manual(1), from file (2) 2
Filename CUBE
 Center (C) & scale (S) data (C,S/N) N
NLM (1), SNLM (2) 2
Are basepoints in a special file (Y/N) N
Basepoints from data file (1), manually (2) 1
 1     0.00E+00  0.00E+00  0.00E+00
 2     0.00E+00  1.00E+00  0.00E+00
 3     1.00E+00  1.00E+00  0.00E+00
 4     1.00E+00  0.00E+00  0.00E+00
 5     0.00E+00  0.00E+00  1.00E+00
 6     0.00E+00  1.00E+00  1.00E+00
 7     1.00E+00  1.00E+00  1.00E+00
 8     1.00E+00  0.00E+00  1.00E+00
Row of basepoint (0 terminates) 1
Row of basepoint (0 terminates) 2
Row of basepoint (0 terminates) 6
Row of basepoint (0 terminates) 0
Converged
```

```
          Distance
 Nspace          2space
-------------------------
 1.0000E+00  9.9886E-01
 1.4142E+00  1.4199E+00
 1.0000E+00  1.0010E+00
-------------------------
Coordinates in 2space
-1.92E-01  1.01E-01
 6.03E-01  7.05E-01
 4.23E-03  1.51E+00

 Transformed 2dim basepoints coordinates
 0.00E+00  0.00E+00
 5.96E-08  9.99E-01
-1.00E+00  1.01E+00

Input mapped points manually(1), existing(2),
 file(3) 2
 Point      X         Y         Error     Code
-----------------------------------------------
 1      0.00E+00  0.00E+00  9.99E-01   bp
 2      5.96E-08  9.99E-01  1.42E+00   bp
 3      9.04E-01  1.23E+00  3.25E-01   fc
 4      7.72E-01  1.58E+00  3.89E-01   nc
 5      0.00E+00  0.00E+00  2.77E-05   ni
 6     -1.00E+00  1.01E+00  1.42E+00   bp
 7      8.69E-01  1.02E+00  3.83E+00   ni
 8      1.19E+00  6.78E-01  7.59E-01   fc
-----------------------------------------------
fc=Final Converged;  nc=Not converged
ni=No Improvement;   bp=Base Point
```

FILE STRUCTURE

The program reads ASCII files with numbers separated by space and in fields 7 columns wide. The first line must contain number of samples (3 columns) number of properties (2 columns) and number of fields per line:

```
 0200606
  1.16    .70   1.50   4.35   4.29    .29        2
   .98   1.05   1.60   4.04   2.16    .27        3
  1.41    .53   1.30   4.85   3.95    .24        4
   .80   1.01   3.40   6.48   4.13    .65        5
.....................................................
        1         2         3         4         5
12345678901234567890123456789012345678901234567890123
.....................................................
```

A basepoint file contains in addition to the matrix layout, the line numbers of basepoints in the original data file:

```
 004060602051218
   .98   1.05   1.60   4.04   2.16    .27        3
  1.08    .98   1.50   8.17   2.82    .25        6
  1.51    .77   1.60   3.74   2.17    .43       13
  1.64   1.00   1.50   3.16   2.20    .30       19
```

Thus for example 004060602051218 means 4 lines (rows) of data (004), in 6 columns (06) and 6 fields. Since the numbers continue, this is a basepoint file and the basepoints are lines 2 (02), 5 (05), 12, and 18 of the original data file.

JOB FILES

For OBRAZJ the format of the JOB file is as follows:

```
!JOB VLADO,USER.CHEMSTRY;TIME=999;INPRI=2
:RUN OBRAZJC
ZFCL                <.........  Input file
ZFCLF               <.........  Output file
S                   <.........  Scale the data
!EOJ
```

The structure of a JOB file for SVETC and the number of data points exceeding 250 is:

```
!JOB VLADO,USER.CHEMSTRY;TIME=600;INPRI=6
:RUN SVETC
N               <.......  Not full printout
2               <.......  Input from file
PKSN            <.......  Input file
Y               <.......  There is a basepoint file
PKS1B           <.......  Basepoint file name
2               <.......  Map points from input file
!EOJ
```

APPLICATIONS

A frequent application deals with a table of numbers, with rows referring to samples and columns to parameters. For example, Leegwater and Leegwater (1981) studied volatile compounds in German and French brandies. Fifteen compounds were measured in 8 samples of each (Table 1).

The nonlinear map produced by SVETC (Fig.5) indicates that one can fairly well separate the two groups of brandies.

The second example deals with mass spectrometry. Mass spectra of 22 hydrocarbons of the formula C9.H12 are in the MSSS data base (CIS, Chemical Information Systems, Inc., 7215 York Road, Baltimore, MD 21212, USA) and are given in Table 2.

The nonlinear map (Fig.6) and the expanded view (Fig.7) indicate the separation of aromatic (1-8) and other hydrocarbons. The simplified nonlinear map (Fig.8), using the compounds 1,5,6 as basepoints also separates some of the substituted benzenes (see for example 2,3 and 4,7) and forms at least two groups out of the other hydrocarbons.

In this example, a 71-dimensional space (m/z range 50-120) was mapped. The program had to be modified to accommodate these arrays. The modification is described in the documentation.

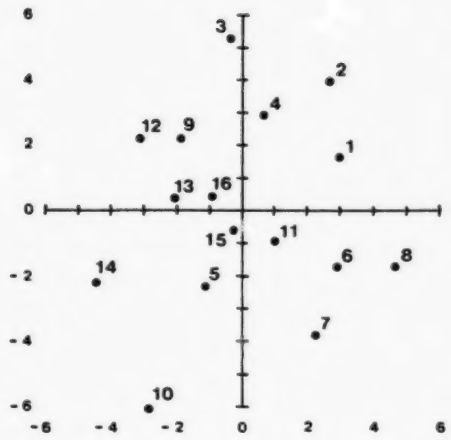


Fig.5. Nonlinear map of flavour components in brandies (Leegwater and Leegwater 1981). The dataset consists of 16 samples (Table 1). The concentrations of 15 components were determined in each sample. German brandies are #1-8; French brandies are #9-16.

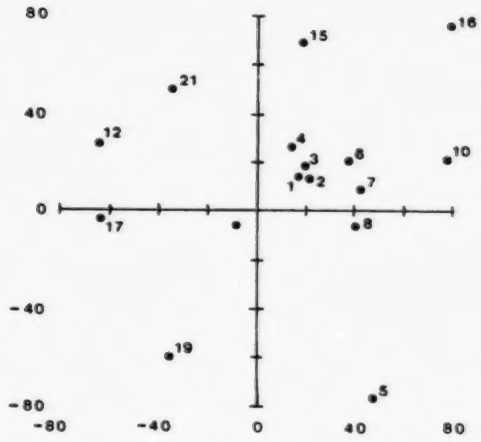


Fig.7. Nonlinear map of mass spectra of hydrocarbons C9.H12. Numbers refer to compounds in Table 2. Expanded view.

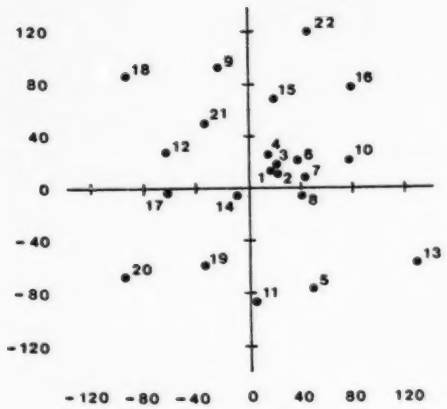


Fig.6. Nonlinear map of mass spectra of hydrocarbons C9.H12. Numbers refer to compounds in Table 2.

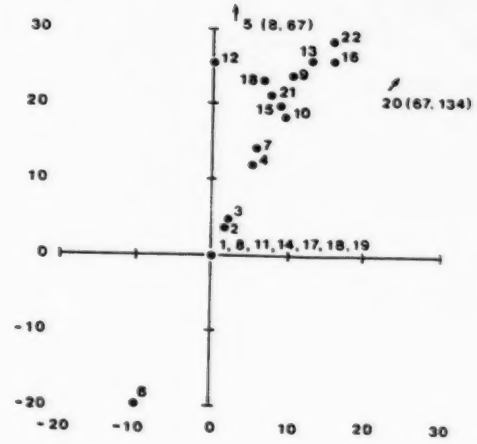


Fig.8. Simplified nonlinear map of mass spectra of hydrocarbons C9.H12. Base points: 1,5,6. Numbers refer to hydrocarbons in Table 2.

ACKNOWLEDGMENTS

I thank Dr.A.W. White and Mr.A. Sreedharan for comments. Mr.S. Bellis wrote a program formatting the manuscript. Messrs. W.G. McMullon and F. Cunningham prepared the figures. Mrs.M.M. Irwin and Ms. Marion Haley proofread the manuscript. Ms. Darlene Warren was the technical editor.

REFERENCES

Anderson, J.W. 1981. Nonlinear mapping. Tektronix Desktop Computer Applications Program Library.

Anon. 1985. ARTHUR Release Version 4.0 User's Manual. Pattern Recognition System. Infometrix, Inc. The Denny Building, 2200 Sixth Avenue, Suite 833 Seattle, WA 98121, V-69.

Derde, M.P. and D.L. Massart. 1982. Extraction of information from large data sets by pattern recognition. Z. Anal. Chem. 313:484-495.

Forina, M., C. Armanino, S. Lanteri, and C. Calcagno. 1983. Simplified non linear mapping of analytical data. Ann. Chim. 73:641-657.

Kowalski, B.R., and C.F. Bender. 1972. Pattern recognition. A powerful approach to interpreting chemical data. J. Am. Chem. Soc. 94:5632-5639.

Kowalski, B.R., and C.F. Bender. 1973. Pattern recognition II. Linear and nonlinear methods for displaying chemical data. J. Am. Chem. Soc. 95:686-693.

Leegwater, D.C., and J.A. Leegwater. 1981. The use of a microcomputer in the classification of grape brandies by pattern recognition. J. Sci. Food Agric. 32:1115-1118.

Massart, D.L., and L. Kaufman. 1983. The Interpretation of Analytical Chemical Data by Use of Cluster Analysis. John Wiley & Sons, New York.

Table 1. Volatile compounds in brandies
(Leegwater and Leegwater 1981)
Concentrations of 15 compounds in 8 samples of German (G1-G8) and French (F1-F8) brandies

0161509									
210	208	1962	97	25	1335	348	4968	439	
10	29	6625	38	1415	139	G1			
417	361	3244	250	22	1711	905	6710	536	
58	31	8825	14	1666	85	G2			
136	984	2551	97	27	1763	199	6895	441	
30	41	8336	24	1870	121	G3			
131	494	2363	177	34	1617	275	7164	544	
81	28	5622	18	1803	100	G4			
30	109	1762	158	36	1375	674	6734	365	
118	14	9575	16	1789	58	G5			
435	79	1376	78	5	1073	808	5241	182	
78	16	7396	16	1368	80	G6			
91	111	739	53	25	788	1262	3087	304	
31	20	3455	12	804	61	G7			
129	341	1323	62	69	1353	4597	4418	301	
42	3	4975	14	1433	60	G8			
96	412	1828	187	1	2301	165	9769	353	
42	26	8899	18	1202	60	F1			
85	100	822	471	544	1749	2160	9283	447	
130	13	13667	20	2014	139	F2			
42	207	962	38	4	1481	23	6368	254	
15	18	5484	21	951	88	F3			
22	429	1708	212	14	2344	59	10896	374	
44	25	11789	19	1217	71	F4			
15	336	1247	103	30	2192	38	8040	341	
26	21	6835	31	999	76	F5			
23	169	1450	146	223	991	1517	6760	146	
62	10	11922	45	2129	121	F6			
21	207	1085	53	12	1648	1625	8110	262	
20	19	7826	22	1423	77	F7			
43	265	1300	165	118	1815	798	8461	311	
49	19	9489	25	1419	90	F8			

Table 2. Mass spectra of C9.H12 hydrocarbons from the MSSS data base of the Chemical Information Systems, Inc.

m/z	Int	m/z	Int	m/z	Int	m/z	Int	m/z	Int
#1	611-14-3	646	120	C9H12					
				Benzene, 1-ethyl-2-methyl- (9CI)					
50	3	51	8	52	2	53	2	58	2
59	2	62	2	63	4	65	6	77	10
78	4	79	7	89	2	91	11	92	3
103	5	104	2	105	100	106	9	115	2
117	2	119	2	120	30				
#2	620-14-4	652	120	C9H12					
				Benzene, 1-ethyl-3-methyl- (9CI)					
50	3	51	7	52	2	53	3	58	2
59	3	62	2	63	5	65	6	77	10
78	4	79	6	89	2	91	11	92	4
103	5	104	2	105	100	106	9	115	2
117	2	119	4	120	32				
#3	622-96-8	631	120	C9H12					
				Benzene, 1-ethyl-4-methyl- (9CI)					
50	3	51	7	52	2	53	2	58	2
59	2	62	2	63	4	65	5	77	10
78	3	79	5	89	2	91	9	92	2
103	5	104	3	105	100	106	9	115	2
117	2	119	3	120	29				
#4	98-82-8	584	120	C9H12					

Benzene, (1-methylethyl)- (9CI)

50	5	51	12	52	3	53	2	63	3
65	2	77	13	78	5	79	10	91	5
103	6	104	2	105	100	106	9	120	25

#5 103-65-1 604 120 C9H12

Benzene, propyl- (8CI9CI)

50	2	51	6	62	2	63	4	65	9
77	3	78	6	91	100	92	10	105	3
120	21								

#6 526-73-8 778 120 C9H12

Benzene, 1,2,3-trimethyl- (8CI9CI)

50	7	51	14	52	5	53	5	58	2
59	2	62	4	63	9	64	2	65	9
74	2	75	2	76	2	77	17	78	7
79	12	89	2	91	14	92	3	102	2
103	8	104	4	105	100	106	9	115	3
117	3	119	11	120	47				

#7 95-63-6 717 120 C9H12

Benzene, 1,2,4-trimethyl- (8CI9CI)

50	5	51	11	52	4	53	5	58	3
62	2	63	6	65	7	75	2	77	15
78	6	79	7	89	2	91	10	92	3
102	2	103	6	104	4	105	100	106	9
115	3	117	3	118	2	119	17	120	56

#8 108-67-8 716 120 C9H12

Benzene, 1,3,5-trimethyl- (9CI)

50	3	51	8	52	3	53	4	57	2
59	3	63	4	65	5	77	13	78	5
79	6	91	9	92	3	102	2	103	6
104	3	105	100	106	9	115	3	117	3
119	15	120	64						

#9 22819-81-4 644 120 C9H12

Bicyclo[3.2.1]oct-2-ene, 3-methylene- (8CI9CI)

50	6	51	14	52	6	53	7	58	2
62	2	63	7	64	2	65	13	66	7
67	2	77	14	78	12	79	26	80	2
90	4	91	100	92	90	93	8	103	3
105	16	106	2	115	2	117	2	119	4
120	31								

#10 17634-51-4 691 120 C9H12

1,3,5-Cycloheptatriene, 7-ethyl- (8CI9CI)

50	7	51	14	52	5	53	4	58	2
62	4	63	9	64	2	65	14	66	2
74	2	75	2	77	15	78	9	79	10
89	4	90	2	91	100	92	12	93	2
102	2	103	8	104	5	105	65	106	6
115	3	117	4	118	2	119	5	120	29

#11 2806-45-3 447 120 C9H12

Cyclohexane, 2-propynylidene- (8CI9CI)

50	16	51	37	52	24	53	25	55	17
59	7	63	20	65	38	66	15	67	22
68	22	73	17	77	37	78	32	79	55
80	8	81	55	91	100	92	63	103	7
105	42	119	7	120	55				

#12 1655-05-6 253 120 C9H12

Cyclohexene, 1-(1-propynyl)- (8CI9CI)

51	9	63	9	65	9	77	40	78	12
79	28	91	91	92	37	103	12	105	95
106	9	119	12	120	100				

#13 55956-43-9 486 120 C9H12

Cyclohexene, 3-(2-propynyl)- (9CI)

50	4	51	8	52	4	53	18	54	2
55	4	63	3	65	5	66	3	67	2

77	11	78	5	79	20	80	4	81	100
82	7	91	10	92	11	105	4	120	2

#14 3141-02-4 553 120 C9H12
1,3-Cyclopentadiene, 5-(1-methylpropylidene)-

50	4	51	17	52	2	53	4	55	7
62	3	63	8	64	2	65	17	66	7
76	3	77	22	78	10	79	22	80	4
89	6	90	4	91	55	92	10	93	3
102	4	103	12	104	5	105	100	106	7
117	4	118	5	119	10	120	61		

#15 41898-89-9 500 120 C9H12
2,3-Heptadien-5-yne, 2,4-dimethyl- (9CI)

50	8	51	18	52	7	53	11	62	5
63	15	64	4	65	28	66	9	67	4
74	4	75	4	77	49	78	11	79	40
80	4	89	4	91	23	92	4	103	18
104	10	105	100	106	9	117	4	119	12
120	73								

#16 2396-65-8 645 120 C9H12
1,8-Nonadiyne (8CI9CI)

50	15	51	25	52	16	53	57	54	7
55	14	62	3	63	6	65	15	66	13
67	10	68	6	77	28	78	20	79	59
80	10	81	18	90	4	91	72	92	31
93	5	103	3	105	28	106	2	119	5

#17 39746-39-9 605 120 C9H12
Spiro[4.4]nona-1,6-diene, (S)- (9CI)

51	7	52	6	53	8	55	10	56	4
63	2	65	14	66	12	67	5	77	31
78	31	79	40	80	6	84	4	89	2
90	3	91	78	92	61	93	10	103	8
104	4	105	100	106	10	115	6	116	2
117	8	118	3	119	18	120	99		

#18 3105-29-1 648 120 C9H12
Tetracyclo[3.3.1.02,8.04,6]nonane (8CI9CI)

50	3	51	11	52	4	53	6	54	2
55	3	62	2	63	5	65	13	66	55
67	5	75	2	77	35	78	61	79	100
80	13	81	2	89	2	90	2	91	64
92	19	93	5	103	5	104	2	105	29
106	2	107	2	108	2	115	4	116	2
117	4	119	6	120	100				

#19 38310-48-4 191 120 C9H12
Tricyclo[3.2.1.02,4]octane, 8-methylene-,

65	14	77	31	78	17	79	55	91	100
92	67	105	35	119	9	120	4		

#20 3048-64-4 503 120 C9H12
Bicyclo[2.2.1]hept-2-ene, 5-ethenyl- (9CI)

50	2	51	4	52	2	53	3	54	2
63	2	65	9	66	100	67	6	77	6
78	3	79	5	91	5	92	2	105	2
120	3								

#21 14296-81-2 784 120 C9H12
Cyclohexane, 1,2,4-tris(methylene)- (8CI9CI)

50	6	51	17	52	9	53	12	54	2
55	3	58	3	59	2	62	2	63	6
64	2	65	13	66	5	67	4	75	2
77	37	78	16	79	47	80	6	89	2
90	2	91	69	92	29	93	6	103	15
104	6	105	100	106	10	115	4	117	5
118	2	119	12	120	79				

#22 16219-75-3 509 120 C9H12
Bicyclo[2.2.1]hept-2-ene, 5-ethylidene- (9CI)

50	15	51	28	52	11	53	14	62	5
63	14	65	21	66	100	67	13	77	24
78	40	79	26	80	2	91	79	92	22
93	3	103	11	105	60	106	4	120	39

APPENDIX

```
OBRAZJ
   10 REM From a program written by Jerry W. Anderson,
   20 REM 12 D1 Phillips Building
   30 REM Bartlesville, OK 74004 (918) 661-7438
   40 REM Tektronix Applications Library Program; Original Date 1/21/81
   50 REM Translated and modified (except for the steepest descent
   55 REM algorithm)
   60 REM                    by VZ 4/9/85.
   65 REM Further modifications 1/11/85 for easier JOB operation
   66 REM and file compatibility with AUTOPLOT
   70 FILES *,*
   80 DIM B0$[72],B$[72],B1$[6]
   90 DIM D[100,10],D1[100,100],C[10],V[10]
  100 DIM C1[100,2],C2[100,2],T[100,2]
  110 DIM N$[8],Y$[1],N1$[8]
  120 DIM X$[50],Y1$[50],T$[50]
  125 O2=0
  126 X$="Mapped X axis"
  127 Y1$="Mapped Y axis"
  130 REM INPUT "Input manual(1), from file (2) ",Q1
  135 Q1=2
  140 IF Q1=2 THEN 260
  150 INPUT "ROWS ",M
  160 INPUT "COLUMNS ",N
  170 FOR I=1 TO M
  180   PRINT "Row ",I
  190   PRINT "---------------"
  200   FOR J=1 TO N
  210     PRINT "Column ",J
  220     INPUT D[I,J]
  230   NEXT J
  240 NEXT I
  250 GOTO 590
  260 INPUT "Input filename ",N$
  270 ASSIGN N$,1,F1
  280 IF F1=0 OR F1=1 THEN 310
  290 PRINT "FILE PROBLEM"
  300 STOP
  310 REM The program writes data in a file compatible with Autoplot
  320 REM It overwrites existing file.
  330 REM The file format is M,X$,Y1$,T$,X,Y
  331 T$=N$
  335 INPUT "Output file ",N1$
  340 CREATE O1,N1$,500
  350 IF O1=1 AND O2=0 THEN DO
  360   PURGE O2,N1$
  370   IF O2=0 THEN 340
  380 DOEND
  381 IF O1=0 THEN DO
  382   ASSIGN N1$,2,O3
  383 DOEND
  390 REM First line must contain R(=M,3digits),C(=N,2 digits),& fields
  400 REM per line (2 digits). The fields are assumed 7 columns wide.
  410 REM Next two digits give number of fields per line
  420 LINPUT #1;B$
  430 B1$=B$[2;3]
  440 CONVERT B1$ TO M
  450 B1$=B$[5;2]
  460 CONVERT B1$ TO N
  470 B1$=B$[7;2]
  480 CONVERT B1$ TO I7
  490 N1=0
  500 ON END #1 THEN 590
  510 LINPUT #1;B$
  520 N1=N1+1
  530 FOR I=1 TO I7
  540   I1=7*I-6
  550   B1$=B$[I1;6]
  560   CONVERT B1$ TO D[N1,I]
  570 NEXT I
  575 IF N1=M THEN 590
  580 GOTO 510
```

```
590 REM D IS THE M * N OBSERVATION MATRIX
600 REM N1 IS THE # OF ITERATIONS
610 REM A IS THE FUDGE FACTOR (.3 TO .4)
620 REM U IS THE # OF DIMENSIONS WE END UP WITH (FOR T)
630 PRINT "        NONLINEAR MAPPING    "
640 REM DATA IDENTIFIERS COULD COME IN HERE
650 REM N1 IS NUMBER OF ITERATIONS
660 N1=40
670 A=.4
680 M1=M-1
690 U=2
700 REDIM D1[M,M],D[M,N],C[N],V[N],T[M,U],C1[M,U],C2[M,U]
710 PRINT "The data matrix is"
720 MAT PRINT D
730 MAT D1=ZER
740 MAT C=ZER
750 MAT V=ZER
760 REM
770 REM U DIMENSION TO WHICH TO REDUCE
780 REM PUT DATA INPUT IN HERE
790 REM
800 INPUT " Center (C) & scale (S) data (C,S/N) ",Y$
810 REM
820 REM
830 REM
840 REM
850 REM
860 REM FIND CENTROID
870 REM IF Y$="N" THEN 1150
880 FOR J=1 TO N
890   S=0
900   FOR I=1 TO M
910     S=S+D[I,J]
920   NEXT I
930   C[J]=S/M
940 NEXT J
950 REM FIND VARIANCE
960 FOR J=1 TO N
970   S=0
980   FOR I=1 TO M
990     S=S+(D[I,J]-C[J])**2
1000  NEXT I
1010  V[J]=S/M
1020 NEXT J
1021 PRINT "Mean"
1022 MAT PRINT C
1023 PRINT "Variance"
1024 MAT PRINT V
1025 IF Y$="N" THEN 1160
1030 REM   STANDARDIZE DATA MATRIX D
1040 FOR P=1 TO M
1050   FOR Q=1 TO N
1060     IF Y$="C" THEN DO
1070       D[P,Q]=D[P,Q]-C[Q]
1080       GOTO 1110
1090     DOEND
1100     D[P,Q]=(D[P,Q]-C[Q])/SQR(V[Q])
1110   NEXT Q
1120 NEXT P
1130 PRINT "The transformed data matrix is "
1140 MAT PRINT D
1150 REM   OBTAIN UPPER TRIANGULAR DISTANCE MATRIX D1
1160 D1=0
1170 FOR I=1 TO M1
1180   I1=I+1
1190   FOR J=I1 TO M
1200     S=0
1210     FOR K=1 TO N
1220       S=S+(D[I,K]-D[J,K])**2
1230     NEXT K
1240     D1[I,J]=SQR(S)
1250   NEXT J
```

```
1260 NEXT I
1270 REM The section 1090-1200 does not seem to serve any purpose
1280 GOTO 1410
1290 B=0
1300 FOR J=1 TO N
1310   IF V[J]<B THEN 1340
1320   I8=J
1330   B=V[J]
1340 NEXT J
1350 B=0
1360 FOR J=1 TO N
1370   IF V[J]<B OR J=I8 THEN 1400
1380   I9=J
1390   B=V[J]
1400 NEXT J
1410 FOR I=1 TO M
1420   FOR W6=1 TO U
1430     T[I,W6]=D[I,W6]
1440   NEXT W6
1450 NEXT I
1460 REM DO NON-LINEAR MAPPING
1470 MAT C1=ZER
1480 MAT C2=ZER
1490 Z=.00001
1500 REM CALCULATE LOWER TRIANGULAR DIST MATRIX FROM INITIAL GUESSES
1510 FOR I=1 TO M1
1520   I1=I+1
1530   FOR J=I1 TO M
1540     W5=0
1550     FOR W4=1 TO U
1560       W5=W5+(T[I,U]-T[J,U])**2
1570     NEXT W4
1580     D1[J,I]=SQR(W5)
1590     IF D1[I,J]>0 THEN 1610
1600     D1[I,J]=Z
1610     IF D1[J,I]>0 THEN 1630
1620     D1[J,I]=Z
1630   NEXT J
1640 NEXT I
1650 REM BEGIN ITERATIVE PROCESS
1660 FOR L=1 TO N1
1670   PRINT L;
1680   FOR I=1 TO M
1690     REM INITIALIZE PARTIALS
1700     FOR W8=1 TO U
1710       C1[I,W8]=0
1720       C2[I,W8]=0
1730     NEXT W8
1740     REM EVALUATE FIRST ORDER PARTIALS
1750     FOR J=1 TO M
1760       IF I=J THEN 1810
1770       D2=SGN(J-I)*(D1[I,J]-D1[J,I])/(D1[I,J]*D1[J,I])
1780       FOR K=1 TO U
1790         C1[I,K]=C1[I,K]+D2*(T[I,K]-T[J,K])
1800       NEXT K
1810     NEXT J
1820     REM EVALUATE 2ND ORDER PARTIALS
1830     FOR J=1 TO M
1840       IF I=J THEN 1960
1850       IF I>=J THEN 1870
1860       A1=D1[J,I]
1870       IF I<=J THEN 1890
1880       A1=D1[I,J]
1890       F1=D1[I,J]*D1[J,I]
1900       F2=(D1[I,J]-D1[J,I])*SGN(J-I)
1910       F4=1+F2/A1
1920       FOR K=1 TO U
1930         F3=(T[I,K]-T[J,K])**2/A1
1940         C2[I,K]=C2[I,K]+(F2-F3*F4)/F1
1950       NEXT K
1960     NEXT J
1970   NEXT I
```

```
1980   REM UPDATE SOLUTION MATRIX T
1990   FOR I=1 TO M
2000     FOR K=1 TO U
2010       T[I,K]=T[I,K]+A*C1[I,K]/ABS(C2[I,K])
2020     NEXT K
2030   NEXT I
2040   REM UPDATE LOWER TRIANGLE OF THE DISTANCE MATRIX D1
2050   FOR I=1 TO M1
2060     I1=I+1
2070     FOR J=I1 TO M
2080       W2=0
2090       FOR W3=1 TO U
2100         W2=W2+(T[I,W3]-T[J,W3])**2
2110       NEXT W3
2120       T9=SQR(W2)
2130       D1[J,I]=T9
2140       IF D1[J,I]>0 THEN 2160
2150       D1[J,I]=Z
2160     NEXT J
2170   NEXT I
2180 NEXT L
2185 PRINT
2190 S=0
2200 E=0
2210 FOR I=1 TO M1
2220   I1=I+1
2230   FOR J=I1 TO M
2240     S=S+D1[I,J]
2250     E=E+(D1[I,J]-D1[J,I])**2/D1[I,J]
2260   NEXT J
2270 NEXT I
2280 E=E/S
2290 REM DELETE C1,C2
2300 MAT C1=ZER
2310 MAT C2=ZER
2320 REM
2330 PRINT "Mapping error ";E
2335 PRINT "The 2-dimensional coordinates are "
2340 MAT PRINT T
2342 Q1$="N"
2345 REM INPUT &
2346 IF Q1$<>"Y" THEN 2370
2350 PRINT "The distances are "
2360 MAT PRINT D1
2370 IF O3=0 THEN DO
2380   PRINT #2;M,X$,Y1$,T$
2390   MAT PRINT #2;T
2410 DOEND
2420 ELSE DO
2430   PRINT "Results not retained in file ";W1$
2440 DOEND
2450 END
```

```
SVETB
   10 FILES *,*,*,*,*
   20 REM This is a nonlinear mapping (NLM) program.
   30 REM It consists of two modules:
   40 REM   1. Nonlinear mapping (NLM)
   50 REM   2. Simplified nonlinear mapping (SNLM)
   60 REM In NLM, points from Nspace (maximum depends on DIM's)
   70 REM are projected into 2space, preserving as much as possible
   80 REM their original Nspace distances. The program has space for
   90 REM 40 points (10-dim) in the NLM mode. Any number of points
  100 REM may be handled in the SNLM mode.
  110 REM In SNLM, up to 10 "basepoints" are selected in Nspace.
  120 REM The basepoints are mapped by NLM. Then, the coordinates
  130 REM of any number of points in 2space are calculated,
  140 REM maintaining their distances from the basepoints in
  150 REM Nspace and in 2space as equal as possible.
  155 REM The DIM's are set for general use. If adjustments
  156 REM are needed, follow instructions below.
  160 DIM B0$[72],B$[72],B1$[6],N$[8],Y1$[1],X$[60],Y$[60],T$[60]
  165 DIM X0[2],X2[2],E1[2],E2[2],D[10],Y2[10,2]
  166 REM The dimensions of the above are fixed
  167 REM In the [M,N] below M*N<=2500
  170 DIM X[250,10],X5[250,10]
  175 REM The number of rows in these must be as above; columns are set
  180 DIM Y[250,2],Y0[250,2],G[250,2]
  185 REM These must be at least M*(M-1)/2 for NLM;
  186 REM 820 = 40 points
  187 DIM X1[820],Y1[820]
  188 REM These must be adjusted to the N desired (X3[.,N])
  190 DIM P[10],C2[10],V[10],X3[10,10]
  195 REM If N>10,disable DO loop 6150-6180!
  200 REM Arrays in NLM: -----------------
  210 REM B0$,B$,B1$ reading ASCII files, N$ file name, Y1$ Y/N
  220 REM X[] original data matrix, X1[] their distances;
  230 REM note that space is provided for only 40 points in NLM!
  240 REM C2[] mean row, V[] standard deviation.
  250 REM Y[] coordinates in 2space, Y1[] distances in 2space
  260 REM Y0[] auxiliary distance in 2space, G[] Jacobian
  270 REM Additional arrays are used in SNLM: ----------------
  280 REM Y[] are the 2-dimensional basepoint coordinates,
  290 REM Y2[] is auxiliary for same,
  300 REM P[] are coordinates of a point, input either manually or
  310 REM from a file; D[] contains calculated distances to the
  320 REM basepoints;
  330 REM as basepoints.
  340 REM X0[] 2space coords, X2,X3[] auxiliary,E1[],E2[] derivatives
  350 MAT X=ZER
  360 MAT X1=ZER
  370 MAT Y=ZER
  380 MAT Y1=ZER
  390 MAT C2=ZER
  400 MAT V=ZER
  410 MAT B=ZER
  420 F1=0
  430 REM Flags for Open file routines
  440 F2=W2=I5=0
  450 W3=2
  460 INPUT "Full printout (Y/N) ",Q2$
  470 INPUT "Input manual(1), from file (2) ",Q1
  480 IF Q1=2 THEN 610
  490 INPUT "ROWS ",M
  500 INPUT "COLUMNS ",N
  510 FOR I=1 TO M
  520   PRINT "Row ",I
  530   PRINT "---------------"
  540   FOR J=1 TO N
  550     PRINT "Column ",J
  560     INPUT X[I,J]
  570   NEXT J
  580 NEXT I
  590 GOTO 680
  600 REM -------------- File input & output -----
```

```
610 F=1
620 GOSUB 5050
630 GOSUB 6060
640 X$="Mapped X axis"
650 Y$="Mapped Y axis"
660 T$=N$
670 GOSUB 6790
680 REM ------------ Initialization---------------
690 M0=M
700 N0=N
710 REM M0,N0 needed in SNLM to check dimensional compatibility
720 A1=0
730 REM A1 is iteration counter, A2 is iteration stop
740 REM A3 is derivative step, A4 is step on the gradient
750 REM A7 is convergence stop
760 A2=20
770 A3=.005
780 A4=.4
790 A7=.0001
800 C1=2
810 REM  F1 flag is used in file input sub
820 REM C1 is the dimension of 2space
830 Q4=1
840 REM Initialized NLM/SNLM flag
850 REM M2 is the number of interpoint distances
860 M2=FNR(M)
870 IF M2>820 THEN DO
880   PRINT "Array too large, only SNLM possible"
890   Q4=2
900   GOTO 930
910 DOEND
920 REDIM X1[M2],Y1[M2]
930 REDIM X[M,N],X5[M,N],C2[N],V[N],Y[M,2],Y0[M,2]
940 REDIM G[M,2]
950 IF Q2$="N" THEN 990
960 PRINT "The data matrix is"
970 MAT PRINT X
980 REM ------------------   Scaling  ---------------
990 IF F2>0 THEN 1300
1000 INPUT " Center (C) & scale (S) data (C,S/N) ",Y1$
1010 IF Y1$="N" THEN 1300
1020 FOR J=1 TO N
1030   S=0
1040   FOR I=1 TO M
1050     S=S+X[I,J]
1060   NEXT I
1070   C2[J]=S/M
1080 NEXT J
1090 REM -----
1100 FOR J=1 TO N
1110   S=0
1120   FOR I=1 TO M
1130     S=S+(X[I,J]-C2[J])**2
1140   NEXT I
1150   V[J]=S/M
1160 NEXT J
1170 REM -------- Center and/or scale data matrix X
1180 FOR I=1 TO M
1190   FOR J=1 TO N
1200     IF Y1$="C" THEN DO
1210       X[I,J]=X[I,J]-C2[J]
1220       GOTO 1250
1230     DOEND
1240     X[I,J]=(X[I,J]-C2[J])/SQR(V[J])
1250   NEXT J
1260 NEXT I
1270 IF Q2$="N" THEN 1320
1280 PRINT "The transformed data matrix is "
1290 MAT PRINT USING "6(MD.2DE,X),/";X
1300 REM ****************************************
1310 REM Splitting point between NLM and SNLM
1320 IF Q4=2 THEN 1350
```

```
1330 INPUT "NLM (1), SNLM (2) ",Q4
1340 IF Q4=1 THEN 1390
1350 MAT X5=X
1360 GOTO 2300
1370 REM ********************************
1380 REM --------- Nspace distances ---------
1390 REM Calculate Nspace distances
1400 GOSUB 1460
1410 REM Initialize 2space coordinates
1420 GOSUB 1570
1430 GOSUB 1650
1440 IF Q4=2 THEN RETURN
1450 END
1460 Z=FND(M,N,X[*,*],X1[*])
1470 IF Q2$="N" THEN 1510
1480 PRINT "Distances:"
1490 MAT PRINT USING "4(MD.4DE,X),/";X1
1500 REM Sum up the multispace distances
1510 X2=0
1520 FOR I=1 TO M2
1530   X2=X2+X1[I]
1540 NEXT I
1550 RETURN
1560 REM ------------------------------------
1570 REM Initialize 2dim coordinates
1580 FOR I=1 TO M
1590   FOR J=1 TO 2
1600     Y[I,J]=X[I,J]
1610   NEXT J
1620 NEXT I
1630 RETURN
1640 REM ------------------------------------
1650 REM Calculate 2dim distances
1660 E8=0
1670 Z=FND(M,C1,Y[*,*],Y1[*])
1680 IF Q2$="N" THEN 1720
1690 REM PRINT "2dim distances"
1700 REM MAT PRINT Y1
1710 REM Calculate mapping error
1720 Z=FNE(M,X1[*],Y1[*],E)
1730 IF Q2$="N" THEN DO
1735   PRINT USING "#,D.4DE,X";E
1736 DOEND
1738 ELSE DO
1740   PRINT "2dim coordinates at error ";E
1750   MAT PRINT USING "2(MD.2DE,X),/";Y
1755 DOEND
1760 IF ABS(E-E8)<A7 THEN DO
1770   PRINT "Converged"
1780   GOSUB 5910
1790   RETURN
1800 DOEND
1810 ELSE DO
1820   E8=E
1830 DOEND
1840 IF A1>A2 THEN DO
1850   PRINT "Stopped by iteration count"
1860   GOSUB 5910
1870   RETURN
1880 DOEND
1890 E0=E
1900 REM --------- Calculate the derivatives -------
1910 FOR I=1 TO M
1920   FOR J=1 TO 2
1930     Y[I,J]=Y[I,J]+A3
1940     Z=FND(M,C1,Y[*,*],Y1[*])
1950     Z=FNE(M,X1[*],Y1[*],E)
1960     G[I,J]=(E-E0)/A3
1970     Y[I,J]=Y[I,J]-A3
1980   NEXT J
1990 NEXT I
2000 REM Save initial 2dim coordinates
```

```
2010 MAT Y0=Y
2020 REM PRINT "jacobian"
2030 REM mat print G
2040 REM PRINT "------------"
2050 REM Set up the 2space coordinates
2060 E9=1E6
2070 FOR A4=.1 TO .9 STEP .1
2080   FOR I=1 TO M
2090     FOR J=1 TO 2
2100       Y[I,J]=Y[I,J]-A4*G[I,J]
2110     NEXT J
2120   NEXT I
2130   Z=FND(M,C1,Y[*,*],Y1[*])
2140   Z=FNE(M,X1[*],Y1[*],E)
2150   REM PRINT A4,E
2160   IF E<E9 THEN DO
2170     A5=A4
2180     E9=E
2190   DOEND
2200   REM Restore the original 2space coordinates
2210   MAT Y=Y0
2220 NEXT A4
2230 REM PRINT "------------"
2240 A1=A1+1
2250 REM PRINT "Minimum at ";A5;E9
2260 MAT G=(A5)*G
2270 MAT Y=Y-G
2280 GOTO 1670
2290 REM ================================================
2300 REM The SNLM module.
2310 REM The algorithm was described by M.Forina, C. Armanino,
2320 REM S. Lanteri, and C. Calcagno. 1983. Simplified nonlinear
2330 REM mapping of analytical data. Ann. Chim. 73:641-657.
2340 REM Up to 10 "basepoints" may be selected. Their distances
2350 REM in 2space are determined by NLM. Distances of points
2360 REM to be mapped from basepoints in Nspace are calculated.
2370 REM Taking one point at a time, its coordinates in 2space
2380 REM are optimized by minimizing the mapping error. Basepoints
2390 REM closer to the point in Nspace carry more weight than distant
2400 REM points.
2410 F9=0
2420 INPUT "Are basepoints in a special file (Y/N) ",Q0$
2430 IF Q0$="N" THEN 2550
2440 F=1
2450 GOSUB 5050
2460 GOSUB 6060
2470 IF N0<>N THEN DO
2480   PRINT "The dimension of basepoints ";N
2490   PRINT "incompatible with the dimension of data ";N0
2500   MAT X=X5
2510   GOTO 2420
2520 DOEND
2530 GOTO 3050
2540 REM ------------------
2550 INPUT "Basepoints selected from data file (1), manually (2) ",Q1
2560 IF Q1=1 THEN DO
2570   I1=0
2580   FOR I=1 TO M
2590     PRINT I;
2600     I1=I1+1
2610     FOR J=1 TO N
2620       PRINT USING "#,MD.2DE,X";X5[I,J]
2630     NEXT J
2640     PRINT
2650     IF I1=20 THEN DO
2660       I1=0
2670       INPUT "Continue (Y/N) ",Q1$
2680       IF Q1$="N" THEN 2710
2690     DOEND
2700   NEXT I
2710   I1=0
2720   INPUT "Row of basepoint (0 terminates) ",I2
```

```
2730   IF I2=0 THEN 2800
2740   I1=I1+1
2750   IF I1=11 THEN 2800
2760   FOR J=1 TO N
2770     X[I1,J]=X5[I2,J]
2780   NEXT J
2790   GOTO 2720
2800   M=I1
2810 DOEND
2820 ELSE DO
2830   INPUT "Basepoints ",M
2840   IF M>10 THEN DO
2850     PRINT "No more than 10 basepoints! "
2860     GOTO 2830
2870   DOEND
2880   INPUT "Dimensions ",N
2890   IF N>10 THEN DO
2900     PRINT "No more than 10space! "
2910     GOTO 2880
2920   DOEND
2930   IF N<>N0 THEN DO
2940     PRINT "Data dimension is ";N0;" !"
2950     GOTO 2880
2960   DOEND
2970   FOR I=1 TO M
2980     FOR J=1 TO N
2990       PRINT "Row ";I,"Column ";J
3000       INPUT X[I,J]
3010     NEXT J
3020   NEXT I
3030 DOEND
3040 REM ------------------------------------
3050 Q1=0
3060 M2=FNR(M)
3070 REDIM X[M,N],X3[M,N],Y[M,2],G[M,2],X1[M2],Y1[M2]
3075 MAT X3=X
3080 REM Calculate Nspace distances
3090 GOSUB 1460
3100 REM Initialize 2space coordinates
3110 GOSUB 1570
3120 REM Optimize 2space coordinates
3130 GOSUB 1650
3140 REM 2space coordinates of basepoints are now in Y[I1,N]
3150 Z=FNU(M,Y[*,*])
3160 PRINT " Transformed 2dim basepoints coordinates"
3170 MAT PRINT USING "2(MD.2DE,X),/";Y
3180 MAT Y2=Y
3190 REM Save unsorted coordinates
3200 REM ------------------------------------------
3210 REM
3220 REDIM P[N],D[M],Y1[M]
3230 REM
3240 REM ------------------------------------------
3250 INPUT "Input mapped points manually(1), existing(2), file(3) ",Q1
3260 IF Q1=1 THEN DO
3270   PRINT "Point coordinates(1E9 terminates)"
3280   FOR I=1 TO N
3290     PRINT I;
3300     INPUT P[I]
3310     IF P[1]=1E9 THEN END
3320   NEXT I
3330   REM Do the mapping
3340   GOSUB 3855
3350   GOSUB 6440
3360   GOTO 3270
3370 DOEND
3380 REM --------------------
3390 IF Q1=2 THEN DO
3400   I3=1
3410   GOSUB 6700
3420   FOR I0=1 TO M0
3430     PRINT I0+I5;
```

```
3440      FOR J=1 TO N
3450        P[J]=X5[IO,J]
3460      NEXT J
3470      GOSUB 3855
3480      GOSUB 6440
3490    NEXT IO
3500    REM There are more records to process
3510    IF F2>0 THEN DO
3520      GOSUB 6940
3530      MO=M
3540      I5=I5+250
3550      GOTO 3420
3560    DOEND
3570    GOSUB 6740
3580    END
3590  DOEND
3600  REM  ----------------------------------------
3610  IF Q1=3 THEN DO
3620    GOSUB 6700
3630    F=5
3640    GOSUB 5050
3650    LINPUT #F;B$
3660    B1$=B$[2;3]
3670    CONVERT B1$ TO M
3680    B1$=B$[5;2]
3690    CONVERT B1$ TO N
3700    B1$=B$[7;2]
3710    CONVERT B1$ TO I7
3720    FOR I=1 TO M
3730      LINPUT #F;B$
3740      FOR I1=1 TO I7
3750        I2=7*I1-6
3760        B1$=B$[I2;6]
3770        CONVERT B1$ TO P[I1]
3780      NEXT I1
3790      GOSUB 3855
3800      GOSUB 6440
3810    NEXT I
3820    GOSUB 6740
3830    END
3840  DOEND
3850  REM -------- Distances to basepoints in Nspace -----
3855  MAT X=X3
3860  FOR I=1 TO M
3870    D[I]=0
3880    FOR J=1 TO N
3890      D[I]=D[I]+(X[I,J]-P[J])**2
3900    NEXT J
3910    D[I]=SQR(D[I])
3912  NEXT I
3915  REM
3916  FOR I=1 TO M
3920    IF D[I]<.000001 THEN DO
3930      E=SQR((D[I]-Y1[I])**2)
3940      B1$="bp"
3950      PRINT USING "3(MD.2DE,X),2X,2A";Y2[I,1],Y2[I,2],E,B1$
3960      IF W3=0 THEN DO
3970        PRINT #4;Y2[I,1],Y2[I,2]
3980      DOEND
3990      GOTO 3490
4000    DOEND
4010  NEXT I
4020  REM PRINT "D"
4030  REM ----------------- Sort ----------------------
4040  REM The matrices D[], X[], and Y[] are sorted in ascending
4050  REM order of D[].
4060  FOR I=1 TO M
4070    FOR J=1 TO M-I
4080      X=D[J]
4090      X1=D[J+1]
4100      IF X<X1 THEN 4230
4110      D[J]=X1
```

```
4120      D[J+1]=X
4130      FOR K=1 TO N
4140        T=X[J,K]
4150        X[J,K]=X[J+1,K]
4160        X[J+1,K]=T
4170      NEXT K
4180      FOR K=1 TO 2
4190        T=Y[J,K]
4200        Y[J,K]=Y[J+1,K]
4210        Y[J+1,K]=T
4220      NEXT K
4230    NEXT J
4240  NEXT I
4250  REM ---------------------------
4260  E0=0
4270  REM E0 error value, A1 error increase & iter. counter
4280  A1=0
4290  REM PRINT "Initial estimate"
4300  FOR I=1 TO 2
4310    REM INPUT X0[I]
4320    X0[I]=P[I]
4330  NEXT I
4340  A1=A1+1
4350  REM PRINT "*** ";A1
4360  REM MAT PRINT X0
4370  IF A1>A2 THEN DO
4380    B1$="nc"
4390    REM PRINT "Not converged for this point"
4400    REM PRINT "Error ";E,"Coordinates ";X0[1],X0[2]
4410    RETURN
4420  DOEND
4430  REM ----------- Calculate distances to basepoints & derivs ---
4440  Z=FNV(M,X0[*],Y[*,*],Y1[*])
4450  REM PRINT "Y"
4460  Z=FNT(M,X0[*],Y1[*],D[*],Y[*,*],E1[*],E)
4470  MAT X2=X0
4480  E0=E
4490  E9=1E30
4500  FOR A4=.05 TO .9 STEP .05
4510    FOR I=1 TO 2
4520      X0[I]=X0[I]-A4*E1[I]
4530    NEXT I
4540    Z=FNV(M,X0[*],Y[*,*],Y1[*])
4550    Z=FNT(M,X0[*],Y1[*],D[*],Y[*,*],E2[*],E)
4560    IF Q2$="N" THEN 4590
4570    REM PRINT "2dim coordinates at error ";E;" and step ";A4
4580    REM MAT PRINT USING "2(MD.2DE,X),/";X0
4590    IF E<E9 THEN DO
4600      A5=A4
4610      E9=E
4620    DOEND
4630    MAT X0=X2
4640  NEXT A4
4650  REM PRINT "Minimum at ";A5;E9
4660  IF E0<=E9 THEN DO
4670    MAT X0=X2
4680    B1$="ni"
4690    E=E0
4700    RETURN
4710  DOEND
4720  MAT E1=(A5)*E1
4730  MAT X0=X0-E1
4740  IF ABS(X0[1])<A7 AND ABS(X0[2])<A7 THEN DO
4750    B1$="nv"
4760    REM PRINT "                          ";X0[1],X0[2]
4770    RETURN
4780  DOEND
4790  IF ABS(E9-E0)<2*A7 THEN DO
4800    B1$="fc"
4810    REM PRINT "Final coordinates ";X0[1],X0[2]
4820    RETURN
4830  DOEND
```

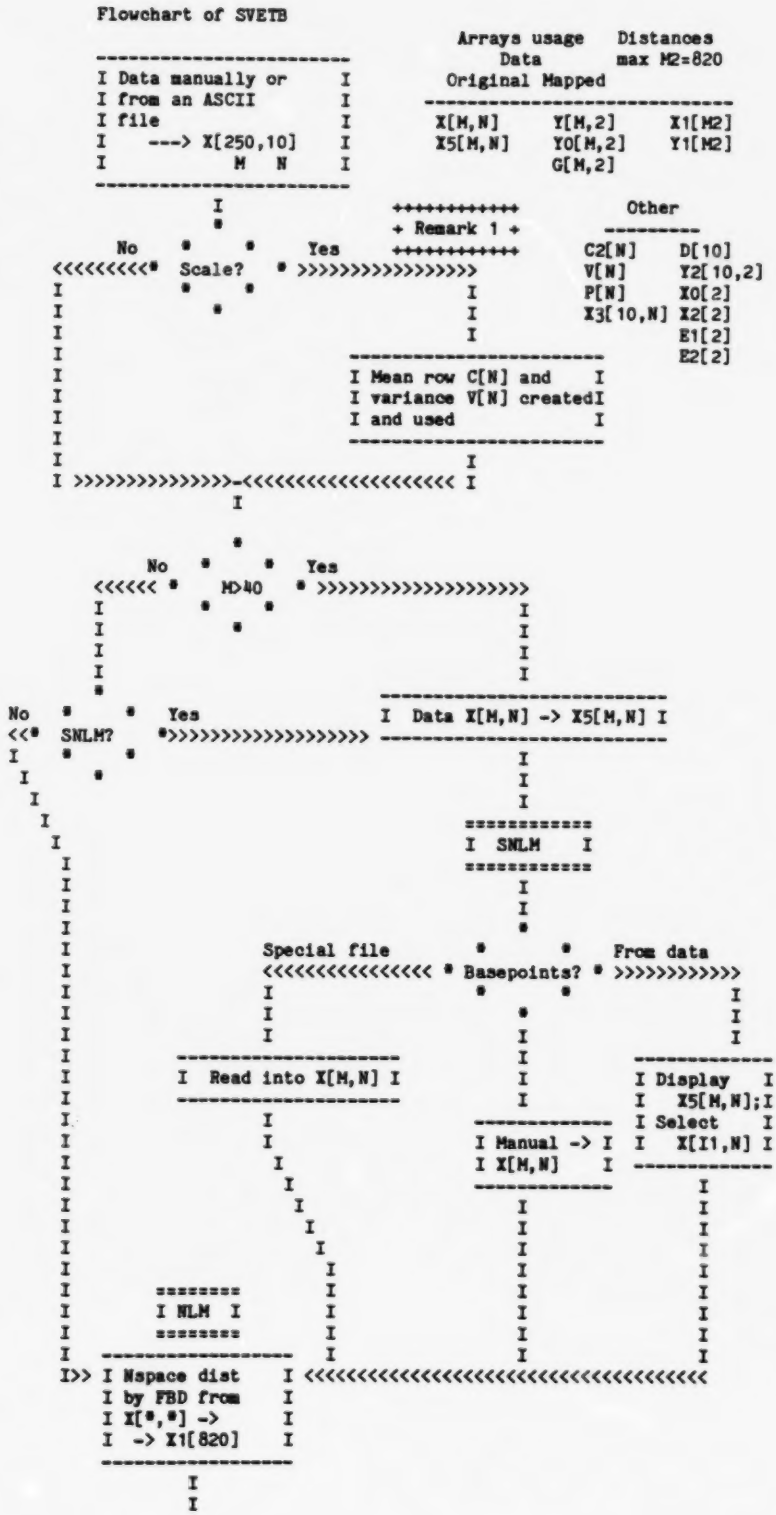
```
4840 GOTO 4340
4850 REM ================================================
4860 REM      Functions definitions & subroutines
4870 DEF FNR(M)=M*(M+1)/2-M
4880 DEF FNV(M,X0[*],Y[*,*],Y1[*])
4890   REM This function calculates 2dim distances from basepoints
4900   REM Input - M-number of points
4910   REM         X0[] - 2dim coordinates of the point
4920   REM         Y[] - 2dim coordinates of basepoints
4930   REM Output -Y1[] - 2dim distances
4940   INTEGER I
4950   MAT Y1=ZER
4960   FOR I=1 TO M
4970     FOR J=1 TO 2
4980       Y1[I]=Y1[I]+(X0[J]-Y[I,J])**2
4990     NEXT J
5000     Y1[I]=SQR(Y1[I])
5010   NEXT I
5020   RETURN Y1[1]
5030 FNEND
5040 REM ------------------------------------------------
5050 REM Open file routine
5060 IF F1=1 THEN 5080
5070 INPUT "Filename ",N$
5080 ASSIGN N$,F,G
5090 IF G=0 OR G=1 THEN 5170
5100 IF G=3 THEN DO
5110   PRINT "File does not exist"
5120   INPUT "Try again (1), quit (2)  ",Q8
5130   IF Q8=1 THEN 5070
5140 DOEND
5150 PRINT "A file problem"
5160 STOP
5170 RETURN
5180 REM ------------------------------------------------
5190 DEF FND(N,C,X[*,*],X1[*])
5200   REM Calculates distances of N points in Cspace
5210   REM X[] and stores them in X1[].
5220   REM Input -  N,C, X[]
5230   REM Output - X1[]
5240   INTEGER I,I1,J,K
5250   REAL D
5260   I1=0
5270   MAT X1=ZER
5280   FOR I=1 TO N-1
5290     FOR J=I+1 TO N
5300       I1=I1+1
5310       FOR K=1 TO C
5320         X1[I1]=X1[I1]+(X[I,K]-X[J,K])**2
5330       NEXT K
5340       X1[I1]=SQR(X1[I1])
5350     NEXT J
5360   NEXT I
5370   RETURN X1[1]
5380 FNEND
5390 REM ------------------------------------------------------
5400 DEF FNE(N,X1[*],Y1[*],E)
5410   REM Calculates sum of squared deviations of arrays
5420   REM X1[] and Y1[]
5430   REM Input X1[], Y1[], N
5440   REM Output Error value E
5450   INTEGER I,I1
5460   I1=N*(N+1)/2-N
5470   E=0
5480   FOR I=1 TO I1
5490     IF X1[I]<.00001 THEN 5510
5500     E=E+(X1[I]-Y1[I])*(X1[I]-Y1[I])/X1[I]
5510   NEXT I
5520   RETURN E
5530 FNEND
5540 REM ------------------------------------------------
5550 DEF FNU(M,Y[*,*])
```

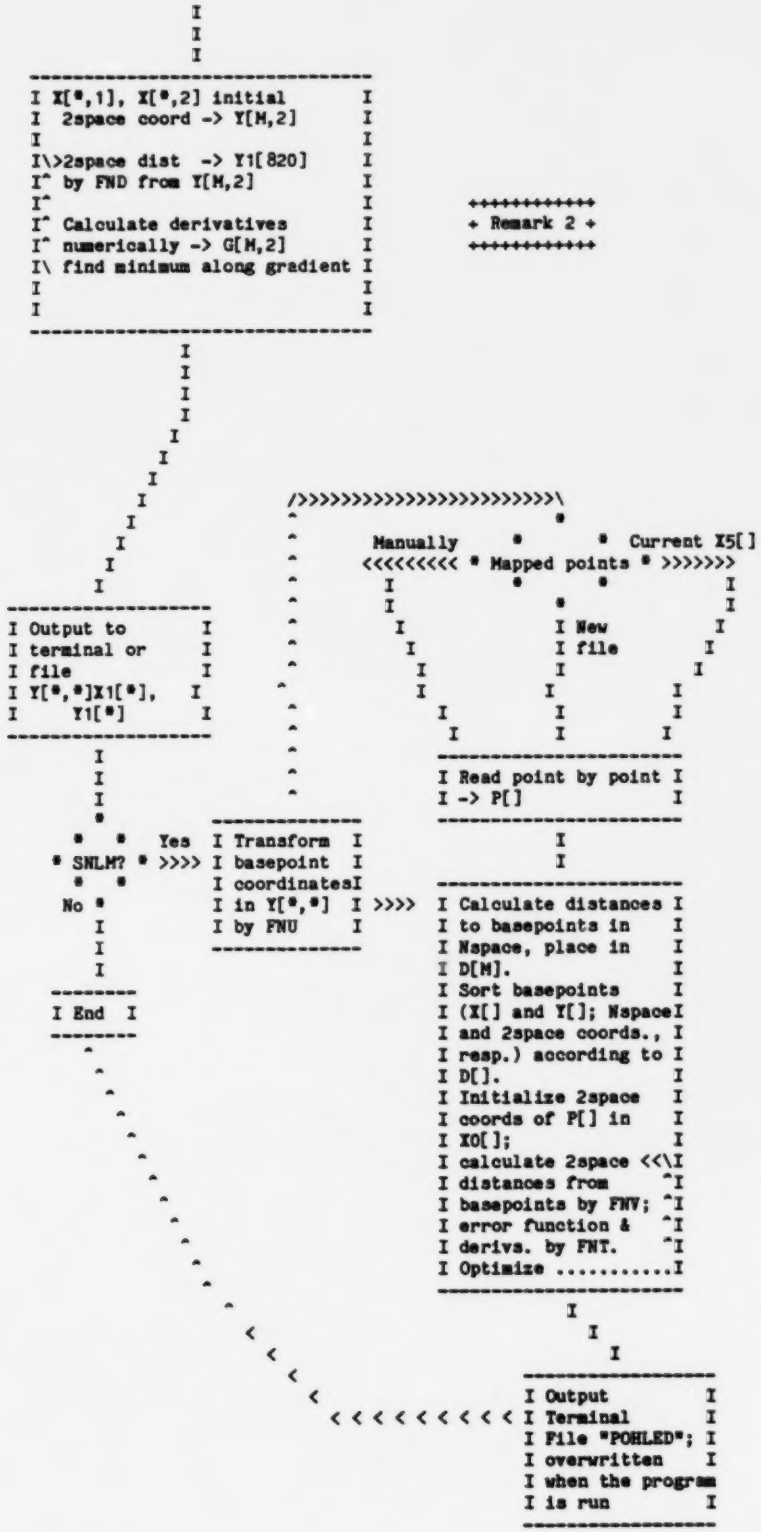
```
5560   REM Matrix Y[M,2] is transformed to have 0's in 1-st row
5570   REM and to have a zero in Y[2,1]. This is accomplished by
5580   REM multiplying by the matrix A5. Note that M<=10!!!
5590   INTEGER I,J
5600   REAL A5[2,2],A3[2,10],A4[10,2],S
5610   REDIM A3[2,M],A4[M,2]
5620   FOR I=2 TO M
5630     FOR J=1 TO 2
5640       Y[I,J]=Y[I,J]-Y[1,J]
5650     NEXT J
5660   NEXT I
5670   Y[1,1]=Y[1,2]=0
5680   S=SQR(Y[2,1]*Y[2,1]+Y[2,2]*Y[2,2])
5690   A5[1,1]=A5[2,2]=Y[2,2]/S
5700   A5[1,2]=-Y[2,1]/S
5705   A5[2,1]=-A5[1,2]
5707   MAT A3=TRN(Y)
5710   MAT A4=A5*A3
5720   MAT Y=TRN(A4)
5730   RETURN Y[1,1]
5740 FNEND
5750 REM ----------------------------
5760 DEF FNS(M,B[*])
5770   REM Sorts array B[M]. Input M, B[], output sorted B[]
5780   INTEGER I,J,K,L
5790   FOR I=1 TO M
5800     FOR J=1 TO M-I
5810       K=B[J]
5820       L=B[J+1]
5830       IF K<L THEN 5860
5840       B[J]=L
5850       B[J+1]=K
5860     NEXT J
5870   NEXT I
5880   RETURN B[1]
5890 FNEND
5900 REM ----------------------------------------------
5910 REM Distances comparison subroutine
5920 PRINT "           Distance "
5930 PRINT " Nspace          2space      Difference"
5940 PRINT "-----------------------------------------"
5950 FOR I=1 TO M2
5955   X9=X1[I]-Y1[I]
5960   PRINT USING "3(MD.4DE,X)";X1[I],Y1[I],X9
5970 NEXT I
5980 PRINT "-----------------------------------------"
5990 PRINT "Coordinates in 2space"
6000 MAT PRINT USING "2(MD.2DE,X),/";Y
6010 IF Q4=1 AND W3=0 THEN DO
6020   MAT PRINT #4;X
6030 DOEND
6040 RETURN
6050 REM ----------------------------------------
6060 REM File input subroutine
6070 REM First line must contain R(=M,3digits),C(=N,2 digits),& fields
6080 REM per line (2 digits). The fields are assumed 7 columns wide.
6090 REM Basepoint file also contains line numbers here
6100 LINPUT #F;B$
6110 B1$=B$[2;3]
6120 CONVERT B1$ TO M
6130 B1$=B$[5;2]
6140 CONVERT B1$ TO N
6150 IF N>10 THEN DO
6160   N=10
6170   PRINT "Only first 10 columns read"
6180 DOEND
6190 B1$=B$[7;2]
6200 CONVERT B1$ TO I7
6210 REM Needed for basepoint files to track bp lines
6220 IF B$[9;2]<>"  " THEN DO
6230   FOR I=0 TO M-1
6240     I1=2*I+9
```

```
6250      B1$=B$[I1;2]
6260      CONVERT B1$ TO B[I+1]
6270    NEXT I
6280  DOEND
6290  REM Point of entry for additional batch of records
6300  IF M>250 THEN DO
6310    F2=M-250
6320    M=250
6330  DOEND
6340  FOR I=1 TO M
6350    LINPUT #F;B$
6355    I3=0
6360    FOR I1=1 TO N
6365      I3=I3+1
6366      IF I3>I7 THEN DO
6367        LINPUT #F;B$
6368        I3=1
6369      DOEND
6370      I2=7*I3-6
6380      B1$=B$[I2;6]
6390      CONVERT B1$ TO X[I,I1]
6400    NEXT I1
6410  NEXT I
6420  RETURN
6430  REM ---------------------------
6440  REM SNLM PRINT routine
6450  PRINT USING "3(MD.2DE,X),2X,2A";X0[1],X0[2],E,B1$
6460  IF W3=0 THEN DO
6470    PRINT #4;X0[1],X0[2]
6480  DOEND
6490  RETURN
6500  REM ----------------------------------------------
6510  DEF FNT(M,X0[*],Y1[*],D[*],Y[*,*],E1[*],E)
6520    REM Calculates SNLM error function E and derivatives
6530    REAL T1,T2,T3,T4
6540    INTEGER I
6550    MAT E1=ZER
6560    E=0
6570    FOR I=1 TO M
6580      IF D[I]=0 OR Y1[I]=0 THEN 6660
6590      E=E+(M+1-I)*((1/D[I]**3)+1/Y1[I]**3)*(Y1[I]-D[I])**2
6600      T1=Y1[I]-D[I]
6610      T2=(2/D[I]**3)+(2/Y1[I]**3)-3*T1/Y1[I]**4
6620      T3=(X0[1]-Y[I,1])/Y1[I]
6630      T4=(X0[2]-Y[I,2])/Y1[I]
6640      E1[1]=E1[1]+(M+1-I)*T1*T2*T3
6650      E1[2]=E1[2]+(M+1-I)*T1*T2*T4
6660    NEXT I
6670    RETURN E1[1]
6680  FNEND
6690  REM Results header
6700  PRINT " Point      X        Y         Error     Code"
6710  PRINT "-----------------------------------------------"
6720  RETURN
6730  REM Results footer
6740  PRINT "-----------------------------------------------"
6750  PRINT "fc=Final Converged;  nc=Not converged"
6760  PRINT "ni=No Improvement;   bp=Base Point"
6770  RETURN
6780  REM ---------------------------------------
6790  REM Basic formatted output
6800  CREATE W1,"POHLED",500
6810  IF W1=1 AND W2=0 THEN DO
6820    PURGE W2,"POHLED"
6830    IF W2=0 THEN 6800
6840  DOEND
6850  IF W1=0 THEN DO
6860    ASSIGN "POHLED",4,W3
6870    IF W3=0 THEN DO
6880      PRINT #4;M,X$,Y$,T$
6890    DOEND
6900  DOEND
```

```
6910 RETURN
6920 REM ------------------------------
6930 REM More records to process
6940 M=F2
6950 F=1
6960 GOSUB 6300
6970 RETURN
```

Flowchart of SVETB




X[*,1], X[*,2] initial
2space coord -> Y[M,2]
2space dist -> Y1[820]
by FND from Y[M,2]
Calculate derivatives
numerically -> G[M,2]
find minimum along gradient
Remark 2
Output to terminal or file
Y[*,*]X1[*], Y1[*]
Manually
Mapped points
Current X5[]
New file
Read point by point
-> P[]
SNLM?
Yes
No
Transform basepoint coordinates in Y[*,*] by FNU
Calculate distances to basepoints in Nspace, place in D[M].
Sort basepoints (X[] and Y[]; Nspace and 2space coords., resp.) according to D[].
Initialize 2space coords of P[] in X0[]; calculate 2space distances from basepoints by FNV; error function & derivs. by FNT.
Optimize .....
End
Output Terminal File "POHLED"; overwritten when the program is run

Defined functions:

```
496      4870 DEF FNR(M)=M*(M+1)/2-M
497      4880 DEF FNV(M,XO[*],Y[*,*],Y1[*])
528      5190 DEF FND(N,C,X[*,*],X1[*])
549      5400 DEF FNE(N,X1[*],Y1[*],E)
564      5550 DEF FNU(M,Y[*,*])
587      5760 DEF FNS(M,B[*])
662      6510 DEF FNT(M,XO[*],Y1[*],D[*],Y[*,*],E1[*],E)
```

Subroutines

NLM distances calculation	[1460]
NLM initialization of 2dim coordinates	[1570]
NLM mapping loop	[1650]
SNLM distances to basepoints in Nspace	[3855]
Open file	[5050]
Distance comparison on NLM conclusion	[5910]
Input file reading	[6060]
2nd entry point (>250 points)	[6300]
SNLM print routine	[6440]
SNLM results header	[6700]
SNLM results footer	[6740]
BASIC file output	[6790]
Auxiliary for >250 records	[6940]

Arrays dimensioning

```
  16    160 DIM BO$[72],B$[72],B1$[6],N$[8],Y1$[1],X$[60],Y$[60],T$[60]
  17    170 DIM X[250,10],X3[10,10],X5[250,10],X1[820],C2[10],V[10]
  18    180 DIM Y[250,2],YO[250,2],Y1[820],Y2[10,2],G[250,2]
  19    190 DIM P[10],D[250],XO[2],X2[2],E1[2],E2[2]
  92    920 REDIM X1[M2],Y1[M2]
  93    930 REDIM X[M,N],X5[M,N],C2[N],V[N],Y[M,2],YO[M,2]
  94    940 REDIM G[M,2]
 311   3070 REDIM X[M,N],X3[M,N],Y[M,2],G[M,2],X1[M2],Y1[M2]
 327   3220 REDIM P[N],D[M],Y1[M]
 570   5610   REDIM A3[2,M],A4[M,2]
```

Remark 1

The DIM's are set for general use. If adjustments
are needed, follow instructions below:
String dimensions are fixed:
DIM BO$[72],B$[72],B1$[6],N$[8],Y1$[1],X$[60],Y$[60],T$[60]
The following arrays are fixed:
DIM XO[2],X2[2],E1[2],E2[2],D[10],Y2[10,2]
In the [M,N] below M*N<=2500:
X[100,15],X5[100,15]
The number of rows in these must be as above; columns are set
Y[100,2],YO[100,2],G[100,2]
These must be at least M*(M-1)/2 for NLM;
820 = 40 points
X1[820],Y1[820]
These must be adjusted to the N desired (X3[.,N])
DIM P[15],C2[15],V[15],X3[10,15]
If N>10,disable DO loop 6150-6180!

Remark 2

The exit from the loop may be by:
1. Decrease of mapping error less than A7 (.0001)
2. No convergence in A2 (20) iterations

Remark 3

The exit from the loop may be by:
1. Decrease of mapping error less than 2*A7

2. No convergence in A2 (20 iterations)
3. No improvement in mapping error
4. Convergence to (0,0) ?

The loop is not entered when the mapped point is a basepoint.